UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/737,129	12/14/2000	Sandra Johnson Baylor	YOR9-2000-0601US1(8728-42	9558

7590 11/12/2004

Frank Chau, Esq.
F. CHAU & ASSOCIATES, LLP
Suite 501
1900 Hempstead Turnpike
East Meadow, NY 11554

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 11/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/737,129	BAYLOR ET AL.	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed August 2, 2004. Claims 1-33 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 103

3. **Claims 1-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Torii (USPN 6,122,712) in view of Steckermeier et al. ("Using Locality Information in Userlevel Scheduling") (hereinafter Steckermeier).**

4. As per claim 1, Torii teaches the invention as claimed, including a method for scheduling threads in a multi-processor computer system having an operating system and at least one cache, comprising the steps of:

storing in a first data structure thread ids for at least some of the threads associated with a context switch performed by the operating system, each of the thread ids uniquely identifying one of the threads (col. 3 lines 47-51); and

storing in a second data structure a plurality of entries for a plurality of groups of contiguous cache lines (col. 4 lines 25-30), each of the plurality of entries arranged such that a thread id in the first data structure is capable of being associated with at least one of the contiguous cache lines in at least one of the plurality of groups of contiguous cache lines (col. 4 lines 31-48), the thread identified by the thread id having accessed the at least one of the

contiguous cache lines in the at least one of the plurality of groups of contiguous cache lines (col. 4 lines 18-26); and

mining for patterns in the plurality of entries in the second data structure to locate multiples of a same thread id that repeat with respect to at least two of the plurality of groups of contiguous cache lines (col. 4 lines 13-18; col. 4 line 66 - col. 5 line 7).

5. Steckermeier teaches the invention as claimed, including scheduling on a same processing unit the threads identified by the located multiples of the same thread id and any other threads identified by any other thread ids associated with the at least two of the plurality of groups of contiguous cache lines (§3; §3.2.2).

6. It would have been obvious to one of ordinary skill in the art to combine Torii and Steckermeier since both consider the problem of scheduling a thread on a processor such that the data the thread accesses is in the cache (Torii, Abstract; Steckermeier, §3). However, Torii only teaches data structures that represent the thread's locality information (col. 3 lines 47-51; col. 4 lines 25-30), while Steckermeier only teaches the scheduling algorithms used for placing the threads on processors (§3.2.2). Thus, both are absent a complete representation of how locality of a thread's data may be used to most efficiently schedule that thread. Steckermeier teaches that threads that operate on the same data should be scheduled on the same processor, yet fails to specifically set forth how such information may be represented in the system (§3; §3.2.2). The combination of Torii and Steckermeier provide a complete model of how to represent data patterns and utilize them to reduce cache misses and increase the performance of a system.

7. As per claim 2, Torii teaches the invention as claimed, including the method according to claim 1, further comprising the step of adding and removing a group to the plurality of groups of contiguous cache lines when a contiguous cache line in the group is accessed by a given thread and when all contiguous cache lines in the group are flushed, respectively (col. 4 line 66 - col. 5 line 7; col. 13 lines 16-23).

8. As per claim 3, Torii teaches the invention as claimed, including the method according to claim 1, further comprising the step of restricting the plurality of groups to a finite number of groups (col. 4 lines 31-34).

9. As per claim 4, Steckermeier teaches the invention as claimed, including the method according to claim 3, further comprising the step of determining when there exists the finite number of groups (§3.4).

10. As per claim 5, Steckermeier teaches the invention as claimed, including the method according to claim 3, wherein said mining step is performed when there exists the finite number of groups (§3.4).

11. As per claim 6, Steckermeier teaches the invention as claimed, including the method according to claim 1, wherein said mining step is performed upon receipt of a command (§3.2.2).

12. As per claim 7, Steckermeier teaches the invention as claimed, including the method according to claim 1, wherein said mining step is performed at least of continuously, at predefined intervals, and upon an occurrence of at least one predefined event (§3.2.2).

13. As per claim 8, Steckermeier teaches the invention as claimed, including the method according to claim 1, wherein said mining step is performed in at least one of software and hardware (§3.2).

14. As per claim 9, Torii teaches the invention as claimed, including the method according to claim 1, wherein said second data structure is comprised of a plurality of rows and a plurality of columns (Fig. 16).

15. As per claim 10, Torii teaches the invention as claimed, including the method according to claim 9, wherein each of the plurality of groups of contiguous cache lines corresponds to one of the plurality of rows (Fig. 16).

16. As per claim 11, Torii teaches the invention as claimed, including the method according to claim 9, wherein each of the thread ids in the second data structure corresponds to one of the plurality of columns (Fig. 16, element 39).

17. As per claim 12, Torii teaches the invention as claimed, including the method according to claim 11, wherein each of the plurality of groups of contiguous cache lines corresponds to one

of the plurality of rows and the any other threads correspond to any of the plurality of columns that intersect any of the plurality of rows corresponding to the at least two of the plurality of groups (Fig. 16, element 39).

18. As per claim 13, Torii teaches the invention as claimed, including the method according to claim 9, further comprising the step of allocating each of the plurality of rows to one of the plurality of groups of contiguous cache lines (Fig. 16, element 18).

19. As per claim 14, Torii teaches the invention as claimed, including the method according to claim 10, further comprising the step of, for each of a cache line in a group in the plurality of groups of contiguous cache lines, storing an index of a row corresponding to the group containing the cache line in the cache line (Fig. 16).

20. As per claim 15, Torii teaches the invention as claimed, including the method according to claim 1, wherein said method is implemented by a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform said method steps (col. 3 lines 25-34).

21. As per claim 16, Torii teaches the invention as claimed, including a method for scheduling threads in a multi-processor computer system having an operating system and at least one cache, comprising the steps of:

storing in a first data structure thread ids for at least some of the threads associated with a context switch performed by the operating system, each of the thread ids uniquely identifying one of the threads (col. 3 lines 47-51); and

storing in a second data structure a plurality of entries for a plurality of groups of contiguous cache lines (col. 4 lines 25-30), each of the plurality of entries arranged such that a thread id in the first data structure is capable of being associated with at least one of the contiguous cache lines in at least one of the plurality of groups of contiguous cache lines (col. 4 lines 31-48), the thread identified by the thread id having accessed the at least one of the contiguous cache lines in the at least one of the plurality of groups of contiguous cache lines (col. 4 lines 18-26); and

mining for patterns in the plurality of entries in the second data structure to locate multiples of a same thread id that repeat with respect to at least two of the plurality of groups of contiguous cache lines (col. 4 lines 13-18; col. 4 line 66 - col. 5 line 7).

22. Steckermeier teaches the invention as claimed, including mapping the threads identified by the located multiples of the same thread id to at least one native thread (§3.2.2).

23. As per claim 17, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein the threads identified by the located multiples of the same thread comprise m threads and the at least one native thread comprises n threads, m and n being integers, m being greater than n (§3.2).

24. As per claim 18, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein said method further comprises the step of scheduling on a same processing unit the threads identified by the located multiples of the same thread id and any other threads identified by any other thread ids associated with the at least two of the plurality of groups of contiguous cache lines (§3.2).

25. As per claim 19, Torii teaches the invention as claimed, including the method according to claim 16, further comprising the step of adding and removing a group to the plurality of groups of contiguous cache lines when a contiguous cache line in the group is accessed by a given thread and when all contiguous cache lines in the group are flushed, respectively (col. 4 line 66 - col. 5 line 7; col. 13 lines 16-23).

26. As per claim 20, Torii teaches the invention as claimed, including the method according to claim 16, further comprising the step of restricting the plurality of groups to a finite number of groups (col. 4 lines 31-34).

27. As per claim 21, Steckermeier teaches the invention as claimed, including the method according to claim 16, The method according to claim 16, further comprising the step of determining when there exists the finite number of groups (§3.4).

28. As per claim 22, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein said mining step is performed when there exists the finite number of groups (§3.4).

29. As per claim 23, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein said mining step is performed upon a receipt of a command (§3.2.2).

30. As per claim 24, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein said mining step is performed at least one of continuously, at predefined intervals, and upon an occurrence of at least one predefined event (§3.2.2).

31. As per claim 25, Steckermeier teaches the invention as claimed, including the method according to claim 16, wherein said mining step is performed in at least one of software and hardware (§3.2).

32. As per claim 26, Torii teaches the invention as claimed, including the method according to claim 16, wherein said second data structure is comprised of a plurality of rows and a plurality of columns (Fig. 16).

33. As per claim 27, Torii teaches the invention as claimed, including the method according to claim 26, wherein each of the plurality of groups of contiguous cache lines corresponds to one of the plurality of rows (Fig. 16).

34. As per claim 28, Torii teaches the invention as claimed, including the method according to claim 26, wherein each of the thread ids in the second data structure corresponds to one of the plurality of columns (Fig. 16, element 39).

35. As per claim 29, Torii teaches the invention as claimed, including the method according to claim 28, wherein each of the plurality of groups of contiguous cache lines corresponds to one of the plurality of rows and the any other threads correspond to any of the plurality of columns that intersect any of the plurality of rows corresponding to the at least two of the plurality of groups (Fig. 16, element 39).

36. As per claim 30, Torii teaches the invention as claimed, including the method according to claim 27, further comprising the step of allocating each of the plurality of rows to one of the plurality of groups of contiguous cache lines (Fig. 16, element 18).

37. As per claim 31, Torii teaches the invention as claimed, including the method according to claim 27, further comprising the step of, for each of a cache line in a group in the plurality of groups of contiguous cache lines, storing an index of a row corresponding to the group containing the cache line in the cache line (Fig. 16).

38. As per claim 32, Torii teaches the invention as claimed, including the method according to claim 16, wherein said method is implemented by a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform said method steps (col. 3 lines 25-34).

39. As per claim 33, Torii teaches the invention as claimed, including a method for scheduling threads in a multi-processor computer system having an operating system and at least one cache, comprising the steps of:

storing in a first data structure thread ids for at least some of the threads associated with a context switch performed by the operating system, each of the thread ids uniquely identifying one of the threads (col. 3 lines 47-51); and

storing in a second data structure a plurality of entries for a plurality of groups of contiguous cache lines (col. 4 lines 25-30), each of the plurality of entries arranged such that a thread id in the first data structure is capable of being associated with at least one of the contiguous cache lines in at least one of the plurality of groups of contiguous cache lines (col. 4 lines 31-48), the thread identified by the thread id having accessed the at least one of the contiguous cache lines in the at least one of the plurality of groups of contiguous cache lines (col. 4 lines 18-26); and

identifying pools of threads in the plurality of entries in the second data structure such that each of the pools of treads comprises the threads identified by a same thread id that forms a multiple with respect to one of the plurality of groups of contiguous cache lines, the multiples

repeating with respect to at least two of the plurality of groups of contiguous cache lines (col. 4 lines 13-18; col. 4 line 66 - col. 5 line 7).

40. Steckermeier teaches the invention as claimed, including scheduling on a same processing unit the threads identified by the located multiples of the same thread id and any other threads identified by any other thread ids associated with the at least two of the plurality of groups of contiguous cache lines (§3.2.2).

Response to Arguments

41. Applicant's arguments filed August 2, 2004 have been fully considered but they are not persuasive.

42. Applicant makes the following arguments relating to the deficiency of Torii:

"*the recited portion of Torii does not teach or suggest 'storing in a first data structure thread ids for at least some of the threads associated with a context switch performed by the operating system.*"

"*Torii does not teach or suggest 'each of the plurality of entries arranged such that a thread id in the first data structure is capable of being associated with at least one of the contiguous cache lines in at least one of the plurality of groups of contiguous cache lines.*"

"*Torii does not teach or suggest 'the thread identified by the thread id having accessed the at least one of the contiguous cache lines in the at least one of the plurality of groups of contiguous cache lines.*"

43. Concerning the first argument, the recited portion of Torii is directed to the thread management unit that manages the generation and elimination of threads. Essentially, this thread management unit acts as a database containing information pertaining to all system threads. The crux of Applicant's argument is that the data associated with these threads is not related to a "context switch". Examiner respectfully disagrees. A context switch, as it relates to multithreading, is simply a processor's changing execution from one thread to another. As the thread management unit stores information relating to all system threads, it follows that the thread management unit would store information relating to threads that are part of a context switch. Regarding the latter two arguments, Torii discusses the cache, which stores data relating to the thread which last accessed it (col. 4 lines 31-48). The cache coherency controller searches through these cache lines to determine if any of the other cache lines have a reference to the thread identifier of the thread that currently holds the thread register. This method of locating cache lines that were accessed by the thread allows the cache coherency controller to identify other cache lines that may be accessed by the thread to reduce the number of miss penalties.

44. Applicant makes the following arguments relating to the deficiency of Steckermeier:

"*Steckermeier is entirely unrelated to mining for patterns...to locate multiples of a same thread id that repeat with respect to at least two of the plurality of groups of contiguous cache lines.*"

"*Steckermeier does not teach or suggest 'mapping the threads identified by the located multiples of the same thread id to at least one native thread.*"

"*Steckermeier does not [t]each or suggest 'identifying pools of threads in the plurality of entries in the second data structure.*"

45. Concerning the first and last arguments, Examiner has remapped the claim limitations and the portions of the references that teach the claim limitations. Steckermeier is no longer relied upon as teaching "mining for patterns" or "identifying pools of threads". Concerning Steckermeier's alleged lacking a teaching of mapping the threads identified to at least one native thread, it is respectfully submitted that the scheduling of threads on a processor inherently meets this limitation. It is well known that native threads is a hardware thread that supports multithreading, and typically exist in a one-to-one relationship with a processor. Essentially, each native thread is a stream of execution on a processor. Thus, when Steckermeier schedules a thread to execute on a processor, the thread has been mapped to the processor, i.e. the native thread.

46. Applicant makes the following arguments relating to the combination of Torii and Steckermeier:

"*The Examiner must explain why the prior art would have suggested to one of ordinary skill in the art the desirability of the modification.*"

47. Examiner has included citations to various sections of Torii and Steckermeier supporting the motivation to combine in paragraph 6. In response to applicant's argument that there is no suggestion to combine the references, the Examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the

references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Steckermeier has clearly set out the advantages to be gained from scheduling a thread on a processor based on cache locality. Clearly, some mechanism is required to represent the thread's cache locality in the system in addition to requiring some method of identifying the locality. Steckermeier's silence on how to achieve these aspects of the scheduling essentially leave the means of achieving this up to the system engineer. Torii provides an exemplary model for representing threads and their cache locality, and it would have been obvious that the representation taught by Torii is one of many possible implementations that would be combinable with Steckermeier.

Conclusion

48. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
November 5, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100